

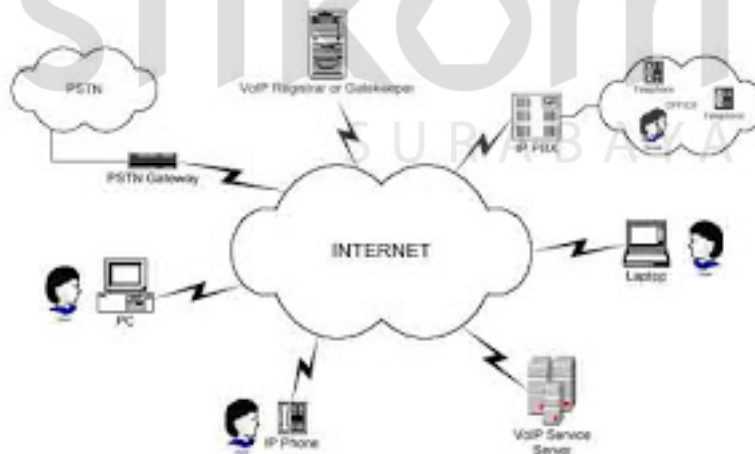
## BAB II

### LANDASAN TEORI

#### 2.1 Voice Over Internet Protocol (VoIP)

*Voice over Internet Protocol* atau yang biasa dikenal dengan sebutan VoIP adalah teknologi yang memungkinkan kemampuan melakukan percakapan telepon dengan menggunakan jalur komunikasi data pada suatu jaringan (*networking*). Sehingga teknologi ini memungkinkan komunikasi suara menggunakan jaringan berbasis IP (*Internet Protocol*) untuk dijalankan diatas infrastruktur jaringan *packet network*. Jaringan yang digunakan bisa berupa internet atau intranet. Teknologi ini bekerja dengan jalan merubah suara menjadi format digital tertentu yang dapat dikirimkan melalui jaringan IP.

### Jaringan VoIP



Sumber: <http://wiralfitasnim91.blogspot.co.id>

Gambar 2.1 Topologi Voip Secara Umum

Tujuan pengimplementasian VoIP adalah untuk menekan biaya operasional dalam melakukan komunikasi. Penekanan biaya itu dapat dilakukan dengan cara memanfaatkan jaringan data yang sudah ada. Sehingga apabila kita ingin membuat jaringan telekomunikasi VoIP tidak perlu membangun infrastruktur baru yang mengeluarkan biaya yang sangat besar. Dengan menggunakan jaringan data yang ada, maka kita melakukan percakapan interlokal maupun internasional hanya dikenakan biaya lokal melalui *Public Switched telephone Network* (PSTN). VoIP lebih mengacu pada layanan komunikasi suara (*voice*), *faksimili*, dan *voice messaging applications*. Teknologi ini pada dasarnya mengkonversi sinyal analog (suara) ke format digital dan kemudian dimampatkan atau ditranslasikan ke dalam paket-paket IP yang kemudian ditransmisikan melalui jaringan. (Budi, Isnanto, & Adhian, 2011).

### 2.1.1 Session Initiation Protokol (SIP)

*Session Initiation Protocol* atau SIP merupakan standar IETF untuk suara atau layanan multimedia melalui jaringan internet. SIP (RFC 2543) di ajukan pada tahun 1999. Pencipta standar ini adalah Henning Schulzrinne. SIP merupakan protokol layer aplikasi yang digunakan untuk manajemen pengaturan panggilan dan pemutusan panggilan. SIP digunakan bersamaan dengan protocol IETF lain seperti SAP, SDP, MGCP (MEGACO) untuk menyediakan layanan VoIP yang lebih luas. Arsitektur SIP mirip dengan arsitektur HTTP (*protocol client-server*). Arsitekturnya terdiri dari *request* yang dikirim dari user SIP ke server SIP. Server itu memproses *request* yang masuk dan memberikan respon kepada client. Permintaan *request* itu, bersama dengan komponen respon pesan yang lain

membuat suatu komunikasi SIP. (Sadewa, 2007)

### 2.1.2 Komponen SIP

Arsitektur SIP terdiri dari dua buah komponen seperti tertera dibawah ini :

- **SIP User Agent**

SIP *User Agent* merupakan akhir dari system (terminal akhir) yang bertindak berdasarkan kehendak dari pemakai. Terdiri dari dua bagian yaitu:

1. *User Agent Client* (UAC) : bagian ini terdapat pada pemakai (*client*), yang digunakan untuk melakukan inisiasi *request* dari server SIP ke UAS.
2. *User Agent Server* (UAS) : bagian ini berfungsi untuk mendengar dan merespon terhadap *request* SIP.

- **SIP Server**

Arsitektur SIP *Server* sendiri menjelaskan jenis-jenis server pada jaringan untuk membantu layanan dan pengaturan panggilan SIP. Diantaranya adalah sebagai berikut :

1. *Registration Server* : server ini menerima *request* registrasi dari user SIP dan melakukan update terhadap lokasi user dengan server.
2. *Proxy Server* : server ini menerima *request* SIP dan meneruskan ke server yang dituju yang memiliki informasi tentang user yang dipanggil.
3. *Redirect Server* : server ini setelah menerima *request* SIP menentukan server yang dituju selanjutnya dan mengembalikan alamat server yang dituju selanjutnya kepada client daripada meneruskan *request* ke server yang dituju tersebut. (Sadewa, 2007)

### 2.1.3 Protokol SIP

SIP menggabungkan beberapa macam protocol baik dari standar yang dikeluarkan oleh IETF maupun ITU – T diantaranya :

1. IETF *Session Description Protocol* (SDP) yang mendefinisikan suatu metode standar dalam menggambarkan karakteristik dari suatu sesi multimedia .
2. IETF *Session Announcement Protocol* (SAP) yang berjalan setiap periode waktu tertentu untuk mengumumkan parameter dari suatu sesi konferensi.
3. IETF *Realtime Transport Protocol* (RTP) dan *Realtime Transport Control Protocol* (RTCP) yang menyediakan informasi tentang manajemen *transport* dan *session*.
4. Algoritma pengkodean. Yang direkomendasikan ITU – T seperti G.733.1, G.729, G728. (Sadewa, 2007)

### 2.2 TrixBos

Trixbos (Asterisk@home) adalah VoIP *Phone System* yang berbasis sistem open source PBX (*Private Branch eXchange*) yang ketika sekali di-*install* kepada suatu PC dengan interface yang sesuai maka akan dapat digunakan sebagai *full feature* PBX untuk pengguna rumahan, lembaga, dan lain sebagainya. Trixbos sangat terkenal karena menggabungkan paket-paket *open source* telepon terbaik yang disertakan di dalam sistem operasi tersebut. Menu utama atau penyetingan Trixbos berbasis web browser untuk menkonfigurasi dan mengatur sistem, serta paket-paket untuk VoIP Server lalu dijadikan dalam satu *bundle* dengan operating system CentOS sehingga menjadi sistem Trixbos CE. (Barrie & Garrison, 2006)



Sumber : <http://tmcnet.com>

Gambar 2.2. Topologi Trixbox

### Komponen TrixBox CE

Komponen yang ada di dalam TrixBox diantaranya :

- **CentOS (Community ENTERprise Operating System)**

CentOS adalah sistem operasi yang dikembangkan oleh komunitas kontributor dan pengguna. Sistem operasi Linux CentOS adalah 100% *rebuild* kompatibel dengan *RedHat Enterprise Linux* (RHEL), dan *full compliance* dengan persyaratan redistribusi RedHat. CentOS ditargetkan untuk siapa saja yang membutuhkan stabilitas tanpa biaya lisensi (*open source*) dan dukungan dari RedHat. (Barrie & Garrison, 2006)

- **Asterisk**

Asterisk adalah sebuah simbol yang merepresentasikan sebuah *wildcard* dibanyak bahasa komputer. Ini merupakan simbol yang menyatakan bahwa Asterisk dikembangkan untuk memenuhi semua tuntutan aplikasi *telephony*. Asterisk dikembangkan dalam lingkungan *Open Source*. Asterisk dirilis oleh para pengembang *open source* dibawah lisensi GNU General Public License,

Asterisk dapat berjalan diatas berbagai macam operating sistem termasuk Linux, Mac OS X, Open BSD, FreeBSD dan Sun Solaris dan menyediakan semua fitur yang akan di harapkan dari sebuah PBX yang sudah termasuk banyak fitur terdepan yang saling berhubungan dengan kelas atas (dengan biaya yang mahal) pada sistem PBX berbayar. Arsitektur Asterisk telah dirancang untuk fleksibilitas yang maksimum dan mendukung semua protokol Voice Over IP. Dan juga bisa dioperasikan dengan semua standar dasar untuk penggunaan peralatan perteleponan yang relatif murah perangkat hardware-nya. (Barrie & Garrison, 2006)

- **Free PBX (Private Branch eXchange)**

Free PBX adalah aplikasi yang digunakan untuk melakukan pengontrolan terhadap jaringan IP telepon private dengan konfigurasi web base, sehingga untuk melakukan konfigurasi terhadap Asterisk tidak diperlukan kemampuan programming karena user interface yang mudah di konfigurasi. (Barrie & Garrison, 2006)

- **Flash Operator Panel (FOP)**

Flash Operator Panel adalah aplikasi yang digunakan untuk melihat status dari semua extensions dan line telepon secara real time. (Barrie & Garrison, 2006)

- **Automated Installation Tools**

Tool yang digunakan untuk sistem operasi, script konfigurasi yang secara otomatis ter-*install* dan ter-*configure* ketika menjalankan Trixbox setup script. (Barrie & Garrison, 2006)

- **Digium Card Auto-Config**

Untuk penambahan Digium Hardware akan secara otomatis terkonfigurasi.  
(Barrie & Garrison, 2006)

### 2.3 Kamailio SIP Server

Kamailio SIP Server merupakan penerus dari OpenSER dan SER yang merupakan *open source* yang *release* dibawah GPL yang bisa menangani ribuan *call setup* per detik. Kamailio bisa digunakan untuk membangun banyak platform untuk VoIP dan komunikasi *realtime* seperti : WebRTC, *Instant messaging* dan beberapa aplikasi lainnya. Fitur yang ditawarkan antara lain: *asynchronous* TCP, UDP dan SCTP, komunikasi aman via TLS untuk VoIP (suara,video,teks); WebSocket support untuk WebRTC; IPv4 dan IPv6; pesan teks; *presence with embedded XCAP server* dan MSRP *relay*; *asynchronous operation*; *IMS extensions* untuk VoLTE, ENUM, DID dan *least cost routing* seperti : *load balancing*, *routing fail-over*, *accounting*, *authentication* dan *authorization*; *support* berbagai macam database seperti MySQL, Postgres, Oracle, Radius, LDAP, Redis, Cassandra, MongoDB, Memcached; *Json* dan *XMLRPC control interface*, *SNMP monitoring*. (kamailio.org)

### 2.4 Linphone

Linphone adalah aplikasi *open source* berbasis *windows*, dan Android yang menawarkan *Voice/Video Call* dan pesan teks secara gratis baik menggunakan server yang telah disediakan oleh Linphone sendiri maupun menggunakan server pribadi, Linphone disini berfungsi sebagai *softphone* yang

menggantikan fungsi IP Phone yang bertugas untuk melakukan panggilan dan menerima panggilan. (linphone.org)

## 2.5 *Quality of Service*

*Quality of Service* (QoS) merupakan mekanisme jaringan yang memungkinkan aplikasi-aplikasi atau layanan dapat beroperasi sesuai dengan yang diharapkan (Thiotrisno, 2011). Kinerja jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah *bandwidth*, *latency* dan *packet loss*, yang dapat membuat efek yang cukup besar bagi banyak aplikasi. Komunikasi suara (seperti VoIP atau *IP Telephony*) serta *video streaming* dapat membuat pengguna frustrasi ketika paket data aplikasi tersebut dialirkan di atas jaringan dengan *bandwidth* yang tidak cukup, dengan *latency* atau *delay* yang tidak dapat diprediksi, atau *packet loss* yang berlebihan.

Dengan menggunakan perhitungan *Quality of Service* (QoS) diharapkan mampu memprediksi *bandwidth*, *latency*, dan *Packet loss* lalu mencocokkannya dengan kebutuhan aplikasi yang digunakan di dalam jaringan tersebut yang tersedia. *Quality of Service* dapat diterapkan pada jaringan melalui mekanisme prioritas pada paket yang masuk ke jaringan, dimana setiap paket yang masuk ke jaringan akan dimonitor terlebih dahulu baik berdasarkan aplikasi maupun *protocol*, kemudian paket-paket mendapatkan prioritas berdasarkan *policy* yang berlaku di jaringan.

QoS dapat dilihat dari tingkat kecepatan dan keandalan dalam mengelola penyampaian data dalam suatu informasi dengan jenis beban yang beragam. Terdapat beberapa parameter yang digunakan untuk mengukur tingkat kecepatan



dan keandalan satu jaringan, diantaranya *latency (delay)*, *packet loss* dan *throughput*.

## 2.6 Parameter–Parameter *Quality of Service*

Pada implementasinya, *Quality of Service* memiliki beberapa parameter yang cukup penting bagi kualitas layanan yang diterima pelanggan, diantaranya:

### 2.6.1 *Delay*

*Delay* adalah waktu yang dibutuhkan oleh sebuah paket data terhitung dari saat pengiriman oleh *transmitter* sampai saat diterima oleh *receiver*. Beberapa jenis *delay* diantaranya adalah sebagai berikut:

- *Propagation delay* (*delay* yang terjadi akibat transmisi melalui jarak antar pengirim dan penerima).
- *Serialization delay* (*delay* yang terjadi saat proses *coding*, *compression*, *decompression* dan *decoding*).
- *Packetization delay* (*delay* yang terjadi saat proses paketisasi *digital voice sample*).
- *Queueing delay* (*delay* akibat waktu tunggu paket sampai dilayani).
- *Jitter buffer* (*delay* akibat adanya *buffer* untuk mengatasi *jitter*) (Thiotrisno, 2011)

### 2.6.2 *Packet Loss*

*Packet loss* adalah banyaknya paket yang hilang selama proses transmisi ke tujuan. *Packet loss* terjadi ketika *peak load* dan *congestion* (kemacetan

transmisi paket akibat padatnya *traffic* yang harus dilayani) dalam batas waktu tertentu, maka *frame* (gabungan data *payload* dan *header* yang ditransmisikan) data akan dibuang sebagaimana perlakuan terhadap *frame* data pada jaringan berbasis IP. (Thiotrisno, 2011)

### 2.6.3 Throughput

*Throughput* merupakan *rate* (kecepatan) transfer data efektif, yang diukur dalam *bit per second* (bps). *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut. (Langi, 2011).

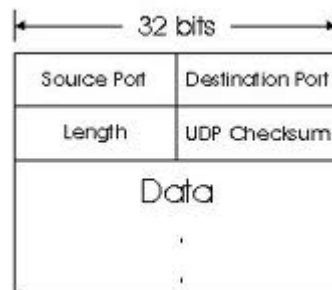
## 2.7 User Datagram Protocol (UDP)

UDP merupakan protocol yang bersifat *connectionless oriented*. Artinya, saat melakukan pengiriman data tidak dilakukan proses *handshaking*, tidak ada *sequencing datagram*, dan tidak ada garansi bahwa paket data (datagram) yang dikirimkan akan tiba dengan selamat, UDP juga tidak menyediakan fitur koreksi kesalahan. (Sofana, 2009)

UDP hanya menyediakan fasilitas *multiplexing* aplikasi (via nomor port) dan integritas verifikasi/deteksi kesalahan (via *checksum*) yang disediakan dalam *header* dan *payload*. Deteksi kesalahan dalam UDP hanya bersifat *optional*. Untuk menghasilkan data yang *reliable*, haruslah dibantu dan dilakukan ditingkat aplikasi. Tidak bisa dikerjakan ditingkat protocol UDP. Pengiriman paket dilakukan berbasis *best effort*.

Data transfer UDP berbeda dengan data transfer pada TCP bahwa tidak

ada penataan kembali. Penggunaan aplikasi UDP mentoleransi terjadinya kehilangan data, atau mempunyai suatu mekanisme untuk mendapatkan kembali data yang hilang.



Sumber : <http://jaringankomputer.org>

Gambar 2.3 Header UDP.

Pada Gambar 2.4 menunjukkan format *header* dari UDP. Delapan *byte* datagram pertama berisi informasi header dan byte tersisa berisi data pesan. Datagram *header* UDP terdiri dari empat bidang dengan masing-masing memiliki ukuran yang sama dengan dua byte (16 bits):

- *Source Port* : memiliki ukuran 16 bits dari 0 sampai 15. Nomor port ini menunjukkan port pengirim.
- *Destination Port* : memiliki ukuran 16 bits. Nomor port ini menunjukkan port yang dituju.
- *Length* : memiliki ukuran 16 bits. Bidang ini menunjukkan panjang total dalam byte seluruh datagram (header + data). Dengan panjang minimum 8 byte, sedangkan panjang maksimum 65,507 byte.
- *Checksum* : Ukuran bidang ini 16 bits juga. Digunakan untuk *error-checking* header dan data. Bidang *checksum* didalam UDP bersifat pilihan. Bila tidak digunakan, *checksum* diset ke nol. Namun harus ditegaskan bahwa *checksum*

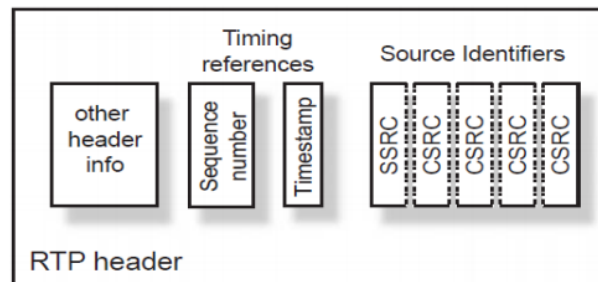
IP hanya digunakan untuk *header* IP dan tidak untuk bidang data, dalam hal ini terdiri dari *header* UDP dan data pemakai. Jadi, bila tidak ada perhitungan *checksum* yang ditampilkan oleh UDP, maka tidak ada pengecekan yang dibuat atas data pemakai.

- Data : Berisikan data.

Overhead yang diperlukan untuk mengirim datagram atau paket UDP sangatlah kecil. Sehingga UDP cocok untuk digunakan pada aplikasi yang membutuhkan *query* dan *response* cepat. Contoh layanan yang cocok untuk UDP yaitu *transmisi audio/video*, seperti: VoIP, *audio/video streaming*. UDP kurang baik jika digunakan untuk mengirim paket berukuran besar. Karena dapat memperlebar peluang jumlah packet loss atau paket hilang. (Sofana, 2009)

## 2.8 Real Time Protocol (RTP)

RTP adalah protokol transportasi yang dikembangkan untuk *streaming* data. RTP berjalan pada (UDP) *User Datagram Protocol*. RTP memiliki *sequence number* dan *timestamp*, yang memfasilitasi waktu data transportasi untuk mengontrol media *server* sehingga proses *stream* dilayani dengan benar untuk ditampilkan secara *real-time*. *Sequence number* digunakan oleh *player* untuk mendeteksi *packet loss* dan mengurutkan paket kedalam urutan yang benar. *Timestamp* adalah pengambilan sampel yang diturunkan dari waktu referensi untuk melakukan sinkronisasi dan perhitungan *jitter* (Austerberry, 2004). *Sequence number* dan *timestamp* terdapat pada bagian RTP *header* seperti terlihat pada Gambar 2.5.



Gambar 2.4. *Header RTP* (Austerberry, 2004)

RTP memiliki bagian data dan bagian *header*. Bagian data seperti *continuous media*, waktu rekonstruksi, *loss detection*, dan identifikasi konten memberikan dukungan untuk aplikasi *real-time*. Bagian *header* terdiri atas segmen IP, segmen UDP, dan segmen RTP (Kurose & Ross, 2010)

## 2.9 Video Protocol 8 (VP 8)

VP8 merupakan *open source* format video kompresi yang didukung oleh perusahaan teknologi, Google. Format VP8 sebenarnya dikembangkan oleh sebuah tim penelitian kecil pada On2 Technologies, Inc sebagai penerus dari keluarga video *codec* VPx. Dibandingkan dengan format video *encoding* lain. VP8 memiliki banyak fitur-fitur teknis khas yang membantu untuk mencapai efisiensi kompresi tinggi maupun rendah untuk proses *decoding* pada waktu yang sama. (Bankoski, Wilkins, & Xu, 2010)

## 2.10 Cloud Computing

*Cloud Computing* adalah gabungan pemanfaatan teknologi komputer 'komputasi' dan pengembangan berbasis Internet 'awan'. Awan (*cloud*) adalah metafora dari internet, sebagaimana awan yang sering digambarkan di diagram

jaringan komputer, awan (*cloud*) dalam *cloud computing* juga merupakan abstraksi dari infrastruktur kompleks yang disembunyikannya. Menurut sebuah makalah tahun 2008 yang dipublikasikan IEEE *Internet Computing Cloud Computing* merupakan suatu paradigma dimana suatu informasi secara permanen tersimpan di *server* (di Internet ) dan tersimpan secara sementara di komputer pengguna (*client*) termasuk di dalamnya adalah *desktop*, komputer tablet, *notebook*, sensor-sensor dan lain lain.

Cloud Computing bisa diartikan sebagai suatu model yang memungkinkan jaringan dapat diakses dengan mudah sesuai kebutuhan di berbagai lokasi dimana model ini memungkinkan untuk mengumpulkan sumber daya komputasi seperti network, server, storage, aplikasi dan services dalam satu wadah.

(Fajrin, 2012)

### 2.11 Wireshark

Wireshark adalah salah satu program tool yang berfungsi sebagai *network analyzer*. Wireshark merupakan program yang menggunakan *Graphical User Interface (GUI)* atau tampilan grafis yang membantu pelanggan untuk mengamati paket di jaringan. Wireshark mampu menangkap semua paket yang melewati jaringan. Paket yang akan diamati pada penelitian ini adalah paket UDP. Setelah paket yang melewati jaringan akan difilter lalu akan dapat mencari pengukuran parameter QoS seperti *delay*, *packet loss* dan *Throughput*.